

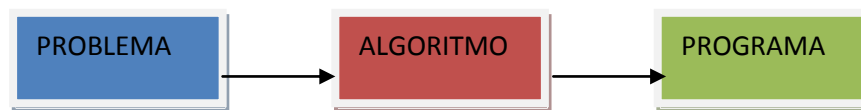
## FLUXOGRAMAS

Para resolver um **problema** através dum computador ou algum sistema microprocessado é necessário encontrar em primeiro lugar uma maneira de descrevê-lo de uma forma clara e precisa. É também preciso que encontremos uma sequência de passos que conduzam à sua resolução. Esta sequência de passos é designada por **algoritmo**. A noção de algoritmo é central para toda solução de algum problema de informática. A criação de algoritmos para resolver os problemas é uma das maiores dificuldades, mas também um dos desafios mais atrativos, dos iniciados em programação.

De um modo geral, considera-se que um algoritmo é uma descrição, passo-a-passo, de uma metodologia que conduz à resolução de um problema ou à execução de uma tarefa. A programação consiste na codificação precisa desse algoritmo, segundo uma linguagem de programação específica.

Se deve levar em consideração que existem três fases distintas na elaboração de programas:

- A análise do problema (especificação do problema, análise de requisitos, pressupostos, etc.)
- A concepção do algoritmo e
- A tradução desse algoritmo na linguagem de programação



### PASSOS NA CONSTRUÇÃO DE ALGORITMOS

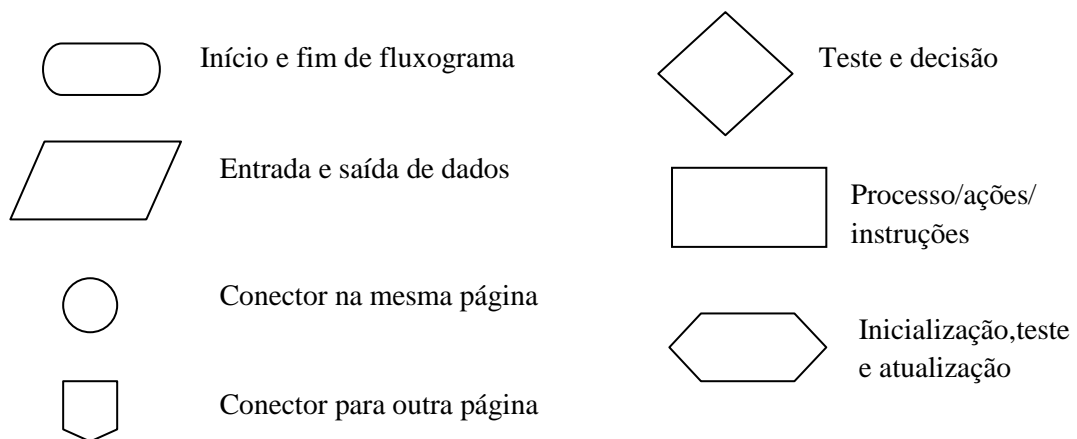
- Compreender o problema
- Identificar os dados de entrada
- Identificar os dados de saída
- Determinar o que é preciso para transformar dados de entrada em dados de saída:
  - ❖ Usar a estratégia do **dividir-para-conquistar**
  - ❖ Observar regras e limitações

- ❖ Identificar todas as ações a realizar
- ❖ Eliminar ambiguidades
- Construir o algoritmo
- Testar o algoritmo
- Executar o algoritmo

Sabemos que uma figura fala por mil palavras. No processo de aprendizado fixamos com mais facilidade imagens do que conceitos escritos.

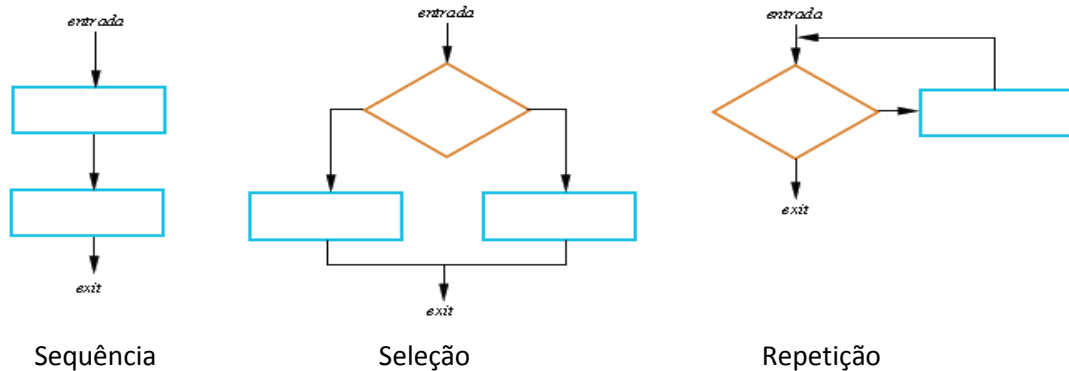
O **diagrama de fluxo ou fluxograma** é uma forma padronizada eficaz para **representar** os passos lógicos de um determinado processamento.

### SIMBOLOGIA EM UM DIAGRAMA DE FLUXO



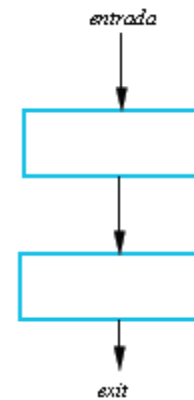
### ESTRUTURAS DE CONTROLE

Uma estrutura de controle é a unidade básica da lógica de programação. Segundo alguns cientistas da computação qualquer programa pode ser construído através da combinação de 3 estruturas básicas: sequência, seleção e repetição.



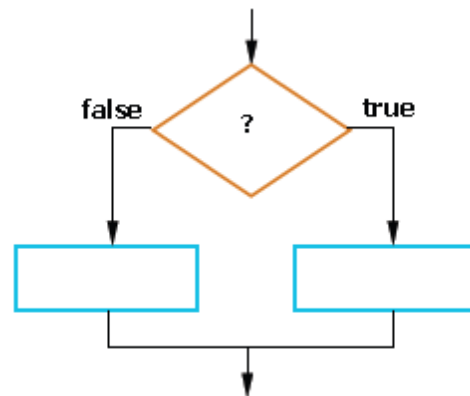
## SEQUÊNCIA

- ❖ Numa sequência é processado um conjunto de ações (ou instruções) em série.
- ❖ Não há qualquer possibilidade de alterar a ordem de processamento das ações, i.e. Após processar a 1ª ação processa-se a 2ª, depois da 2ª processa-se a 3ª, e assim por diante até processar a última ação.
- ❖ Em C, uma sequência é um bloco de instruções que começa com { e termina com }.



## SELECÇÃO COM DUAS VIAS

- ❖ Uma estrutura de seleção é também designada por **estrutura de decisão**.
- ❖ Neste caso, o fluxo de processamento segue por 1 das 2 vias, dependendo do valor lógico (verdadeiro ou falso) da expressão avaliada no início da estrutura.
- ❖ Se o fluxo de processamento só passa por 1 via, então só uma das ações é realizada ou processada.
- ❖ Em C, uma estrutura de seleção com 2 vias é a instrução **if-else**.



```

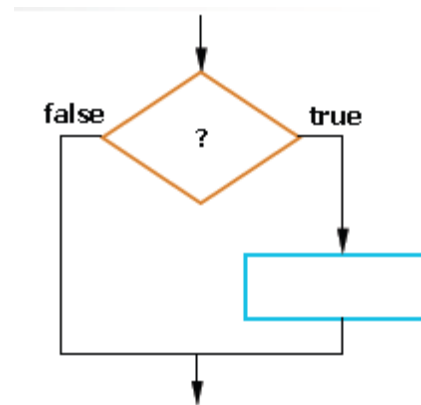
#include <stdio.h>

void main()
{
    int x, y, maior;

    scanf("%d%d\n", &x, &y);
    if (x > y)
        maior = x;
    else
        maior = y;
    printf("O maior dos dois inteiros = %d\n", maior);
}
  
```

## SELEÇÃO COM UMA VIA

- ❖ Neste caso, se a expressão lógica tiver resultado false, nenhuma ação é processada dentro da estrutura de seleção.
- ❖ Só é processada uma ação dentro da estrutura de seleção se a expressão lógica for true; daí, o nome de seleção com 1 via.
- ❖ Em C, uma estrutura de seleção com 1 via é a instrução **if**.



```

#include<stdio.h>

void main()
{
    int nota;

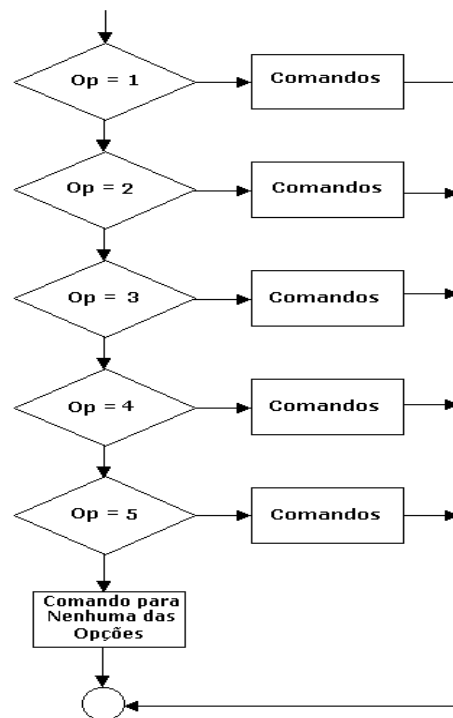
    printf("Introduza o valor da nota: ");
    scanf("%f",&nota);

    if ( (nota >= 9.0) && (nota < 9.5) )
        nota = 9.5;

    printf("A nota = %f\n", nota);
}
  
```

## SELEÇÃO COM N - VIAS

- ❖ Neste caso, a decisão não é feita com base numa expressão lógica porque há mais do que 2 resultados possíveis.
- ❖ Também só são processadas a ação ou as ações encontradas numa via.
- ❖ Em C, uma estrutura de seleção com n vias é a instrução switch com break. No entanto, se não usarmos o break, há a possibilidade de executar as ações de várias vias.



```
#include <stdio.h>

void main()
{
    int count;

    scanf("%d",&count);

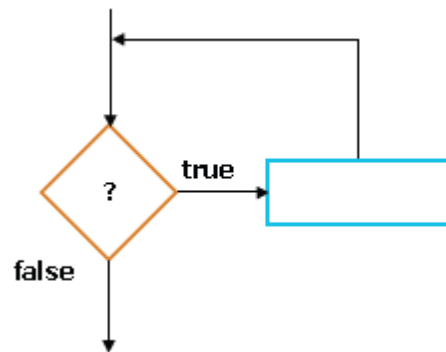
    switch count
    {
        case 5: printf("5\n"); break;
        case 4: printf("4\n"); break;
        case 3: printf("3\n"); break;
        case 2: printf("2\n"); break;
        case 1: printf("1\n");
    }
}
```

## REPETIÇÃO

Utilizamos os comandos de repetição quando desejamos que um determinado conjunto de instruções ou comandos sejam executados um número definido ou indefinido de vezes, ou enquanto um determinado estado de coisas prevalecer ou até que seja alcançado.

### REPETIÇÃO COM TESTE NA CABECA

- ❖ Neste caso, também há a necessidade de tomar uma decisão com base no valor lógico duma expressão.
- ❖ No entanto, a mesma ação será executada repetidamente enquanto o resultado da expressão lógica se mantiver verdadeiro (true).
- ❖ O teste (da expressão lógica) precede a ação.  
Diz-se, por isso, que o teste é na cabeça.
- ❖ O teste é importante porque funciona como uma condição de parada (a false) dos ciclos ou repetições.
- ❖ Em C, uma estrutura de repetição deste tipo é a instrução **while**.



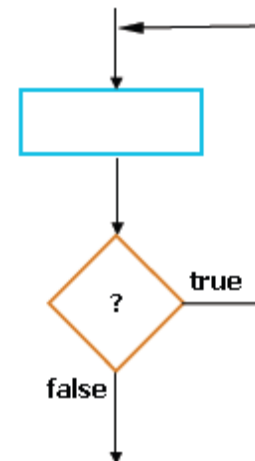
```
#include <stdio.h>

void main()
{
    int sum, k=1;

    sum = 0; // inicializacao da soma
    while (k <=100)
    {
        sum = sum + k;
        k = k + 1; // atualiza a variavel do ciclo
    };
    printf("A soma = %i\n", sum);
}
```

## REPETIÇÃO COM TESTE NA CAUDA

- ❖ Esta estrutura de repetição é em tudo idêntica à anterior. A diferença é que o teste é feito após o processamento da ação.
- ❖ O teste (da expressão lógica) sucede a ação. Diz-se, por isso, que o teste é na cauda.
- ❖ Em C, uma estrutura de repetição deste tipo é a instrução **do-while**.



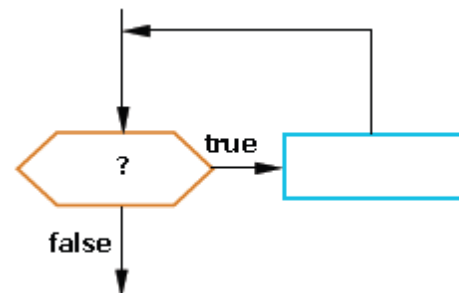
```

#include <stdio.h>
main()
{
    int sum, k=1;

    sum = 0;           // inicializacao da variavel do ciclo
    do
    {
        sum = sum + k;
        k = k + 1;     //actualiza a variavel do ciclo
    }
    while (k<=100);
    printf("A soma =%i\n", sum);
}
  
```

## REPETIÇÃO COM NÚMERO PRE-DEFINIDO DE CICLOS

- ❖ Esta estrutura de repetição é em tudo idêntica às anteriores.
- ❖ O teste é feito na cabeça.
- ❖ A diferença é que é logo na partida é especificado o número de ciclos (ou iterações) que serão efetuados, i.e. o número de vezes que a ação será processada.



- ❖ Em C, uma estrutura de repetição deste tipo é a instrução **for**.

```
#include <stdio.h>

void main()
{
    int    i;           // variavel do ciclo
    int    sum;

    sum = 0;           // inicializacao da soma
    for (i=1; i <=100; i++)
    {
        sum = sum + i;
        printf("A soma = %i\n", sum);
    }
}
```

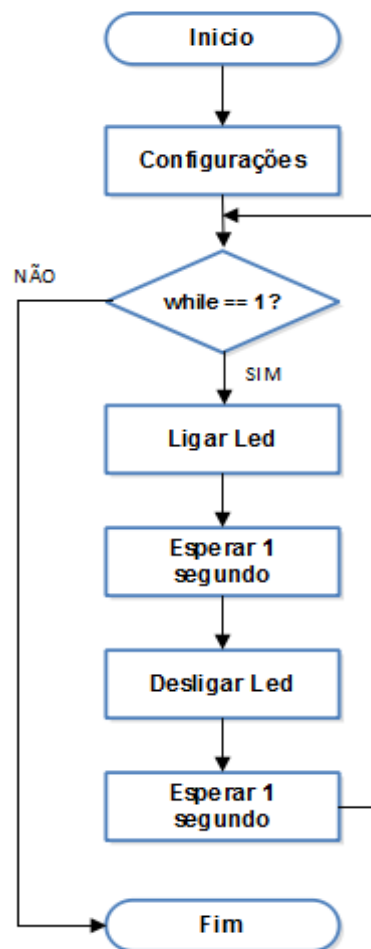
inicialização

teste de paragem

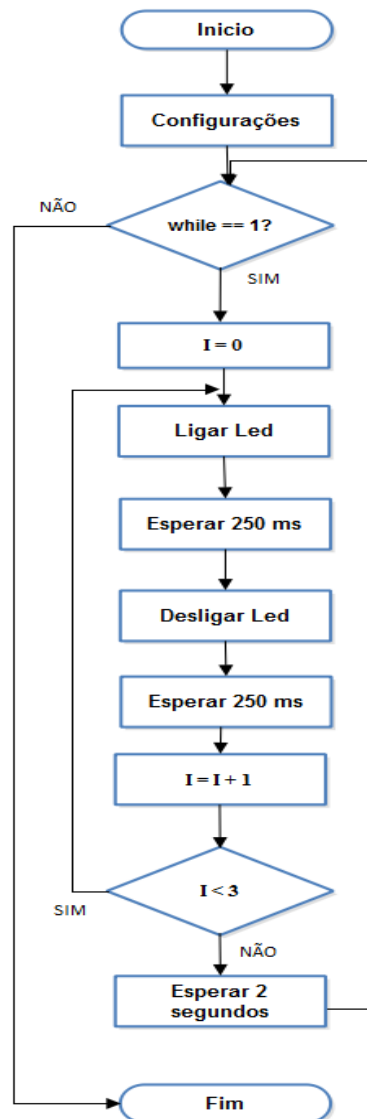
actualização

## EXEMPLOS

1. Um led é conectado a um pino da porta B do microcontrolador. O led pisca continuamente com um intervalo de 1 segundo. Fazer o fluxograma deste projeto.



2. Um led está conectado a um pino do microcontrolador. O led está piscando continuamente segundo a seguinte sequência: 3 piscadas com 250 ms de intervalo entre cada liga-desliga e depois 2 segundos de atraso indefinidamente. Fazer o fluxograma deste projeto.

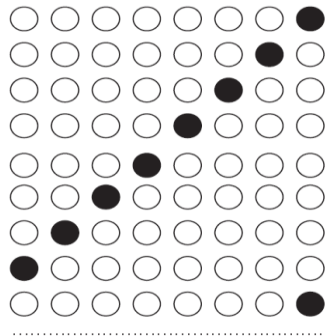


3. Neste projeto, 8 leds são conectados à porta B do microcontrolador. Quando o projeto começa (ou quando é resetado), os leds contam em binário com um atraso de 250 ms entre cada conta. A conta vai de 0 a 255 e se repete indefinidamente. Fazer o fluxograma deste projeto.
4. Neste projeto, 8 leds são conectados à porta B do microcontrolador. Quando o projeto começa (ou quando é resetado), os leds se deslocam para esquerda com um atraso de 250 ms entre cada saída como mostra a figura. Quando o led mais a



esquerda está ligado, o próximo led a ser ligado deve ser o led mais a direita (bit 0).

Este processo se repete indefinidamente. Fazer o fluxograma deste projeto.



5. Repetir o exercício 4 só que o deslocamento é para direita. Fazer o fluxograma deste projeto.
6. Neste projeto, 8 leds são conectados à porta B do microcontrolador e uma tecla está conectada a outro pino do microcontrolador. Normalmente os leds se deslocam para esquerda, mas quando se pressiona a tecla os leds se deslocam para direita. Fazer o fluxograma deste projeto.